Sandia National Laboratories

Exceptional service in the national interest

ME469: A Verification and Validation (V&V) Methodology

Stefan P. Domino[1,2]

[1] Computational Thermal and Fluid Mechanics, Sandia National Laboratories

[2] Institute for Computational and Mathematical Engineering, Stanford

SAND2018-4536 PE

U.S. DEPARTMENT OF ENERGY

NNSA

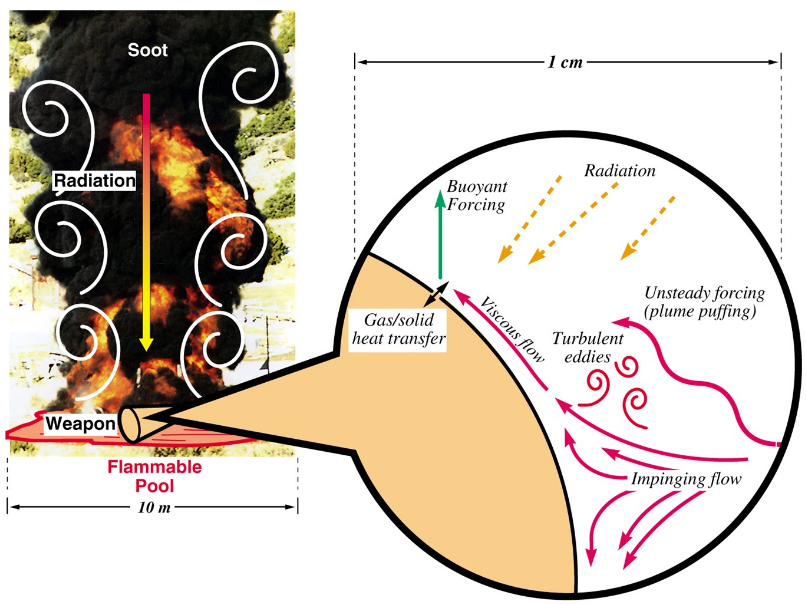
Verification and Validation Objectives

Introduce and Discuss:

- Elements of a Poor/High-Quality Verification Study
- Elements of a Pool/High-Quality Validation Study
- Overview of a Phenomena Identification Ranking Table (PIRT)
  - Cast within a fire safety use case
- Solution Verification
- The Role of Code Verification, including the Method of Manufactured Solutions (MMS)
- Code Verification: Value Proposition
- Review

Our Exemplar: Heat Transfer Mechanisms: Abnormal/Thermal Environment (ATE)

- Characterized by a highly sooting, turbulent reacting flow with participating media radiation (PMR) and conjugate heat transfer (CHT) *multi-physics* coupling
- Liquid fuels, e.g., JP-8, are common due to its usage in transportation, however, propellants (point-Lagrangian/Eulerian) and composite (heterogeneous combustion) also encompass the ATE




Propellant upward burn

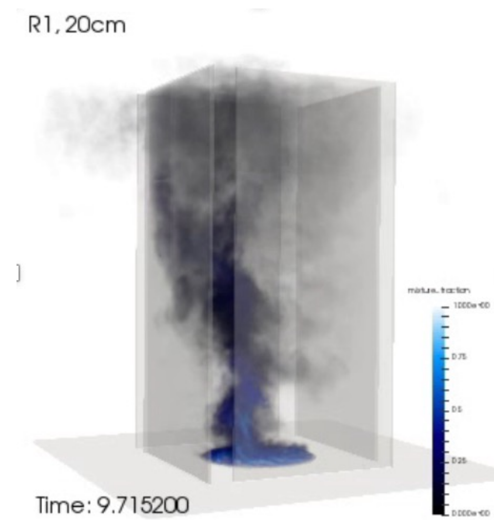

Anderson Air Force B-2 Accident (2/2008)

[1] https://en.wikipedia.org/wiki/2008_Andersen_Air_Force_Base_B-2_accident

Objective: Deploy a High-Fidelity Tool to Predict a Fire with External Flow Coupling

Consider that we want to develop and deploy a foundational understanding in, e.g., fire physics and deploy a simulation tool to support production analysis

- Let us introduce the fire-environment with elements of the following physics:
  - Chemical reactions, soot, buoyancy, turbulent fluid mechanics, thermal radiation heat transfer, convective heat transfer, object heat-up/response, and sometimes propellants
  - As mixing increases, accurately capturing convective heat loads becomes important
  - This perspective and definition defines our ”conceptual” model



Volume-rendered mixture fraction

Verification vs Validation (V&V)? The Formal Lexicon (REVIEW)

<u>Verification</u>: Are we solving the equations correctly?

- Represents an exercise in computational mathematics
- Given an equation, is the solution converging at known rates?

<u>Validation</u>: Are we solving the correct equations?

- Represents an exercise in understanding the physics associated with the real world use case

In this course, we will strongly focus on *verification*

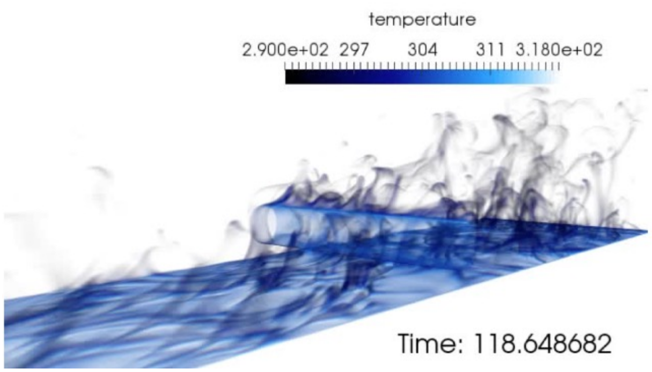
- Establishing the correctness of the numerical implementation is key
- Comparisons of the numerical results to reality is not the primary objective

Verification challenges?

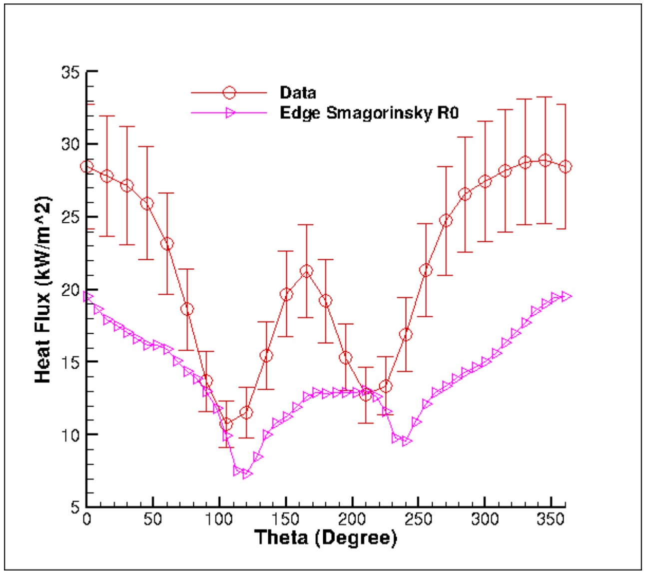
- Knowledge of the true solution, i.e., exact analytical solutions
- How many exact solutions exist for our class of physics? Not many!
- Hint: Method of Manufactured Solutions (MMS)

Challenge: Understanding Errors….

- One mesh, one model, unknown code/numerical pedigree…
- We need to distinguish the types of errors:
  - Model-form Error
  - Discretization Error
  - Code Error?



Heat flux to the cylinder
Volume-rendered temperature



Time-averaged heat flux to cylinder

- What credible scientific hypothesis can be tested in this context?

Does a Process Exist?
Phenomenological Identification and Ranking Table (PIRT)

A Phenomenological Identification and Ranking Table (PIRT) is:

- A process that defines 1) what you know, 2) what you think you know, and 3) what "you know not of"
- Legacy to the United States Nuclear Regulatory Commission (NRC), "*The Phenomena Identification and Ranking Table (PIRT) process is a systematic way of gathering information from experts on a specific concept, and ranking the importance of the information, in order to meet some decision-making objective*" [1]
- Extensively used in the planning of high-consequence mod/sim/exp plans

[1] "Quantifying Reactor Safety Margins: Application of CSAU to a LBLOCA," B.E. Boyack, et al, Part 1: An overview of the CSAU Evaluation Methodology; G.E. Wilson et al., Part 2: Characterization of Important Contributors to Uncertainty, W. Wulff et al., Part 3: Assessment and Ranging Parameters; C.S. Lellouche et al., Part 4: Uncertainty Evaluation of analysis Based on TRAC-PF1/MOD1; N. Zuber et al., Part 5: Evaluation of Scale-up Capabilities of Best Estimate Codes; I. Catton et al., Part 6: A Physically Based Method of Estimating PWR LBLOCA PCT, Nuclear Engineering and Design 119 (1) pp 1-117, May 1990.

Step 1: Phenomena Identification and Ranking Table (PIRT)

Identify all salient physics and rate the following:

- The importance of the phenomenon
- Adequacy of Models
- Adequacy of Code (does the code have the required models; has the code been “verified”)
- Adequacy of Validation (do we believe that the models capture the physical world)

Once a particular use-case is understood and a PIRT is complete, one can proceed down the path of simply closing the gaps in the modeling, code implementation, and model validation process

Example: Convective heat transfer:

	Import	Adequacy			
	Phen	Mod	Code	Val	Mats
Convective Processes					
Convective heat transfer	M	M	M	L	

- We believe that we have a good understanding of the models, and the code seems to have the models that we want implemented (given our current understanding)
- However, high-quality validation experimental data is missing
- Investment: Experimental campaign for mixed-convective heat transfer

Review of the Method of Manufactured Solutions (MMS):

Providing confidence that the code implementation converges to the proper solution

- We understand that the number of analytical solutions to test our code implementation are very few in number
- How can we test the numerical accuracy of our implementation that, in general, solves very complex physics?
- Specifically, as we refine the mesh and time step, how does the error respond?

Consider a simple heat conduction PDE:

$$\rho C_p \frac{\partial T}{\partial t} - \frac{\partial}{\partial x_j} \lambda \frac{\partial T}{\partial x_j} = 0$$

With given [steady] manufactured solution:

$$T^{mms}(x, y, z) = \frac{k}{4\lambda} \left(cos(2\pi x) + cos(2\pi y) + cos(2\pi z) \right)$$

- New, analytically modified system that includes a new source term that we can implement in the code base:

$$\rho C_p \frac{\partial T^{mms}}{\partial t} - \frac{\partial}{\partial x_j} \lambda \frac{\partial T^{mms}}{\partial x_j} = S^{mms}$$

- The error is computed to be the difference between the analytical, or manufactured solution and our numerical simulation, T^h
- We can now refine the mesh and timestep, while computing the error to ensure that the rate of reduction is expected
- For example, if we believe our scheme is 2^{nd} or 3^{rd} order in space accuracy, one uniform refinement should reduce the error by 4x or 8x, respectively

Review of the Method of Manufactured Solutions (MMS):
Providing confidence that the code implementation converges to the proper solution

- We understand that the number of analytical solutions to test our code implementation are very few in number
- How can we test the numerical accuracy of our implementation that, in general, solves very complex physics?
- Specifically, as we refine the mesh and time step, how does the error respond?

Consider a simple heat conduction PDE:

$$\rho C_p \frac{\partial T}{\partial t} - \frac{\partial}{\partial x_j} \lambda \frac{\partial T}{\partial x_j} = 0$$

With given [steady] manufactured solution:

$$T^{mms}(x,y,z) = \frac{k}{4\lambda}\left(cos(2\pi x) + cos(2\pi y) + cos(2\pi z)\right)$$

- New, analytically modified system that includes a new source term that we can implement in the code base:

$$\rho C_p \frac{\partial T^{mms}}{\partial t} - \frac{\partial}{\partial x_j} \lambda \frac{\partial T^{mms}}{\partial x_j} = S^{mms}$$

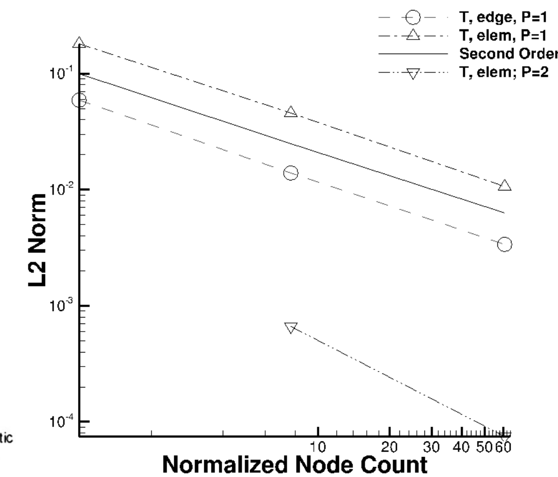
- The error is computed to be the difference between the analytical, or manufactured solution and our numerical simulation, T^h
- We can now refine the mesh and timestep, while computing the error to ensure that the rate of reduction is expected
- For example, if we believe our scheme is 2^{nd} or 3^{rd} order in space accuracy, one uniform refinement should reduce the error by 4x or 8x, respectively

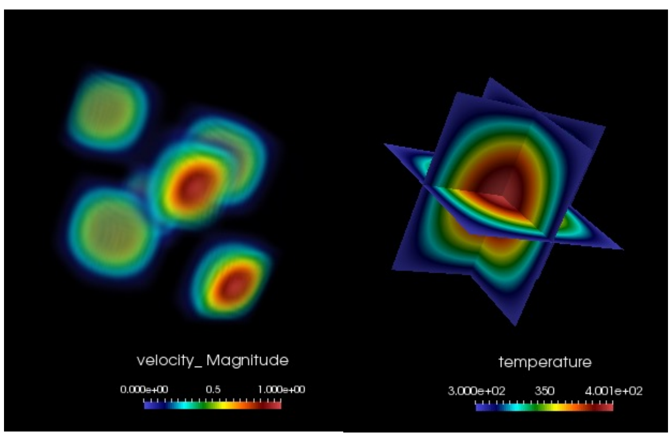
$$S^{mms}(x,y,z) = k\pi^2\left(cos(2\pi x) + cos(2\pi y) + cos(2\pi z)\right)$$

Spatial Code Verification for a low-Mach, Variable-Density Flow

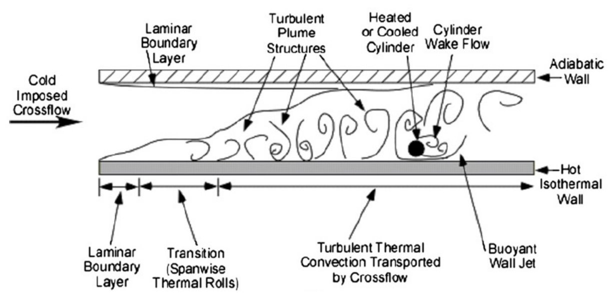
	Import	Adequacy			
	Phen	Mod	Code	Val	Mats
Convective Processes					
Convective heat transfer	M	M	M	L	

- Density is a function of static enthalpy transport via the standard ideal gas, $\rho = f(P,M,R,T)$
- Temperature range maps to experiment (see below)
- Arbitrary buoyancy source term via rotated gravity vector
- Collective study now provides confidence in the interplay between numerical and modeling accuracy





Velocity Mag Temperature



Kearney experimental configuration

See, "Exploring model-form uncertainties in large-eddy simulations", Domino et al, 2016

Solution Verification

Wait, we have a verified code base, however, are running a physics that, while maps to the verification problem, does not have a known solution….

- Solution Verification: The process by which we establish that the numerical error associated with a simulation study is *quantified*
- We may be seeking a quantity of interest such, e.g., heat flux to an object, a velocity profile that is of interest, pressure field, drag coefficient, etc.

Ideally, as we refine the mesh, we will see a *reduced* error

- However, an error relative to what? An experiment? Another code result? The other simulations?

Formally, the product of a solution verification study is a convergence rate for a quantity of interest

Richardson Extrapolation: Obtain a series of simulations (that are sufficiently resolved) that are combined to extrapolate to a higher-order solution estimate (given an order of accuracy assumption)

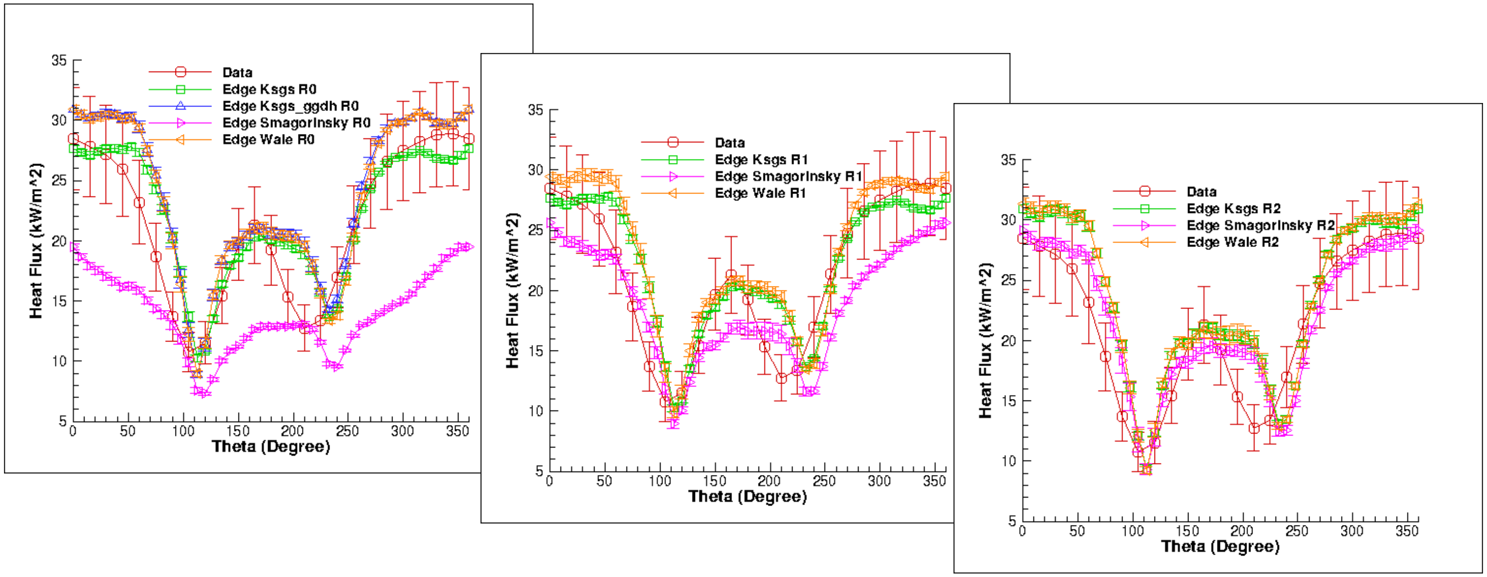
- See: Roache, P.J. and Knupp, P.M. (1993), Completed Richardson extrapolation. Commun. Numer. Meth. Engng., 9: 365-374. https://doi.org/10.1002/cnm.1640090502

In practice, most quality studies meet the concept of "solution sensitivity", i.e., how does the solution vary based on the refined mesh spacing and time step?

Solution Sensitivity & Due Diligence for Model-form (Structural) Uncertainty

- Three meshes (solution sensitivity)
- Three models (structural uncertainty)

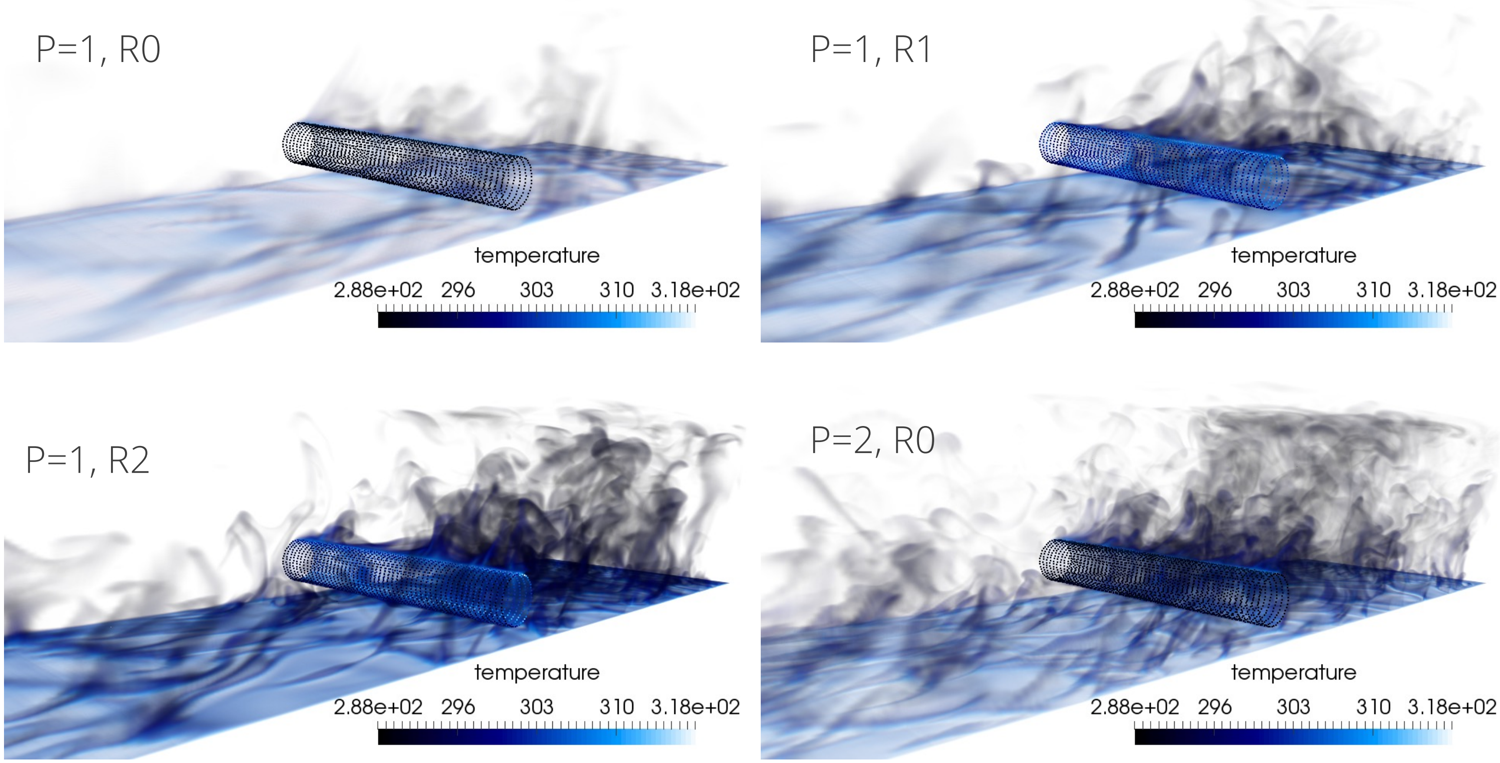
The heat flux results also show error bars due to time and spatial averaging over a line-of-site



- Can we yet feel confidence in any scientific statement? (much closer)

The Effect of Polynomial (P) and Mesh Spacing (H) Refinement

For the heated cylinder-in-cross flow, either mesh resolution or polynomial promotion provides a more accurate result with more turbulence structure noted



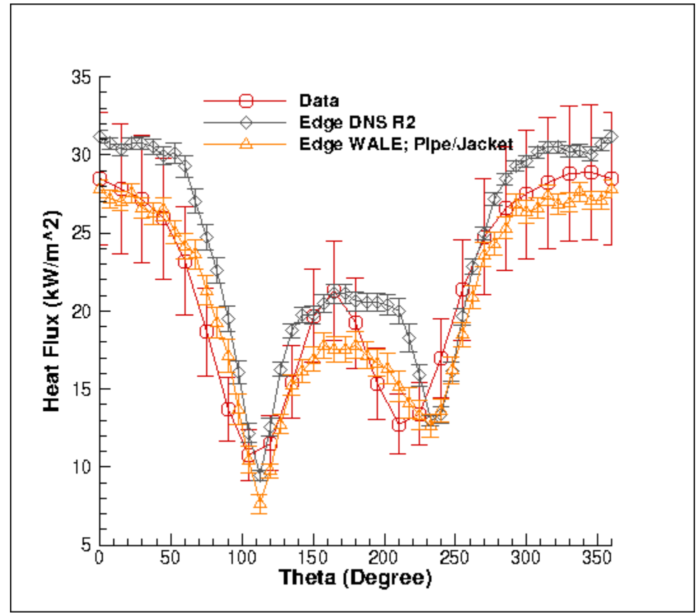
Review Case and Conceptual Model:
Why are we not converging to the experimental result?

- Model Configuration: SNL-based Sean Kearney Experiment, “Experimental investigation of a cylinder in turbulent convection with an imposed shear flow”, *AIAA*, 2005
- Experiment: Outer crossflow, *outer cylinder wall*, inner hot fluid
- QoI: Heat flux… How was that obtained?
  - Inverse analysis! Experiment measured temperature and inferred heat flux; an additional error/uncertainty

<u>Iteration on the conceptual model</u>? Add two fluid regions <u>*and*</u> a CHT coupling to remove the usage of removed the conceptual error an usage of the inverse wall temperature…

Still not perfect, however, we have included many forms of uncertainty in the simulation study

- Inflow velocity, wall temperature, etc. are more options

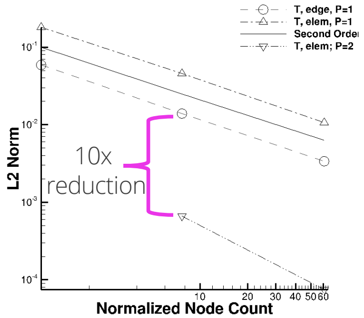


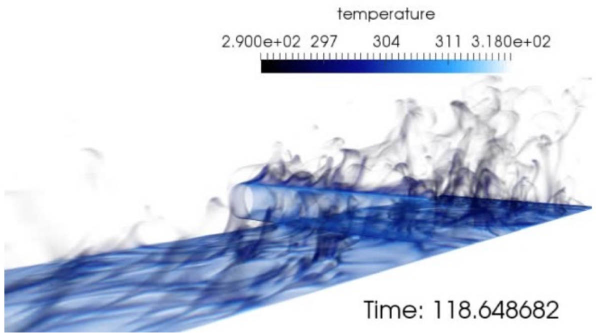
Review of a Strong V&V Process

Established a sound LES-based V&V process (with uncertainty quantification) that includes the following attributes:

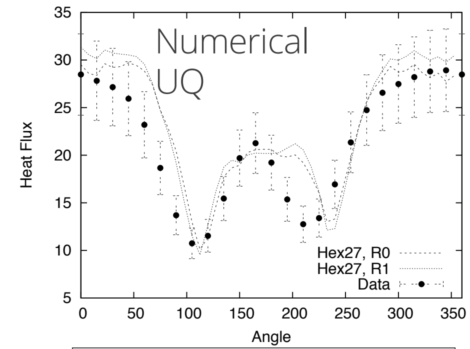
- Definition of key physics, PIRT
- Code verification
- Validation including solution sensitivity
- Structural uncertainty, i.e., model form, quantification
- <u>Physics assumptions</u>, i.e., your conceptual model

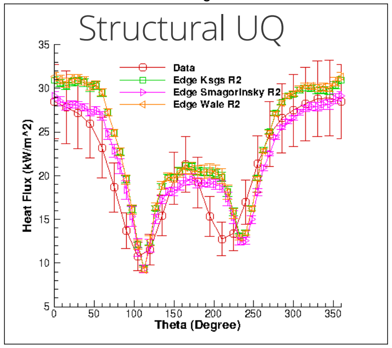
Non-isothermal MMS

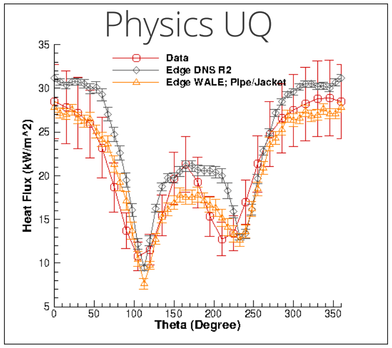




"An assessment of atypical mesh topologies for low-Mach LES", Domino et al., *Comp & Fluids*, 2019







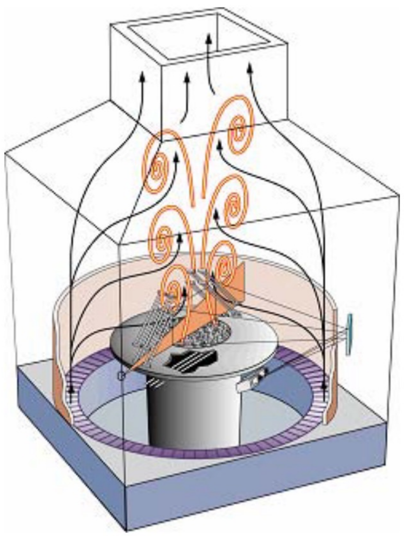


Figure 6. Schematic of the Flame Facility Showing Relationship of Plume, Laser Illumination, and Cameras.

Tieszen et al. “Validation of a Simple Turbulence Model Suitable for Closure of Temporally-Filtered Navier-Stokes Equations Using a Helium Plume”, SAND2005-3210

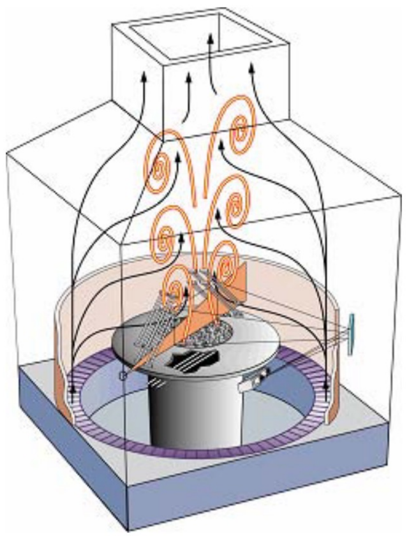
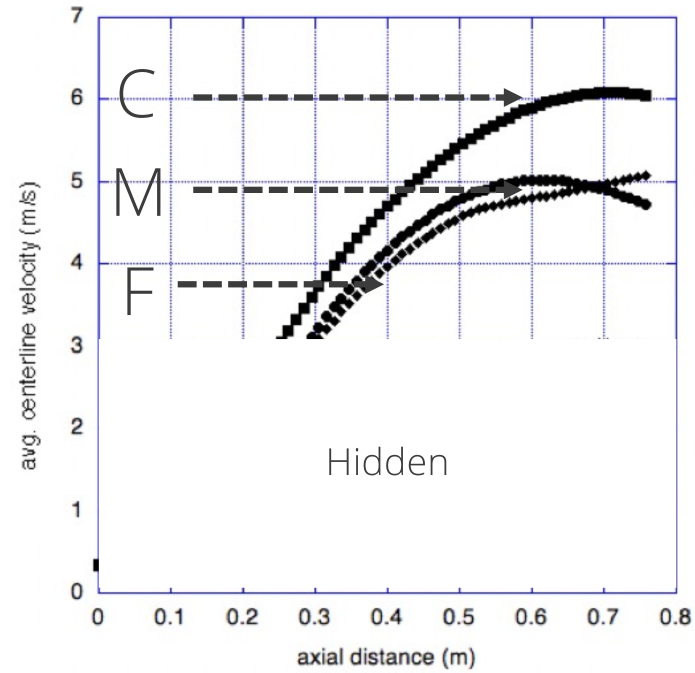


Figure 6. Schematic of the Flame Facility Showing Relationship of Plume, Laser Illumination, and Cameras.



Tieszen et al. “Validation of a Simple Turbulence Model Suitable for Closure of Temporally-Filtered Navier-Stokes Equations Using a Helium Plume”, SAND2005-3210

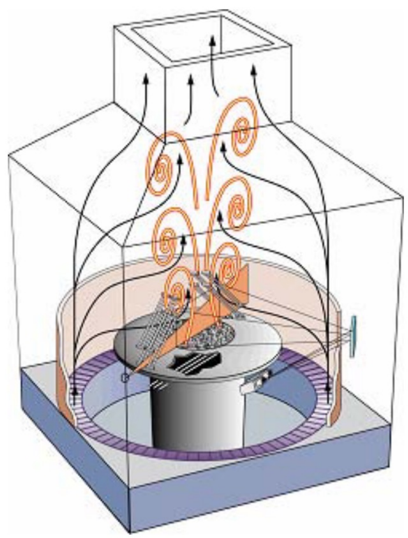
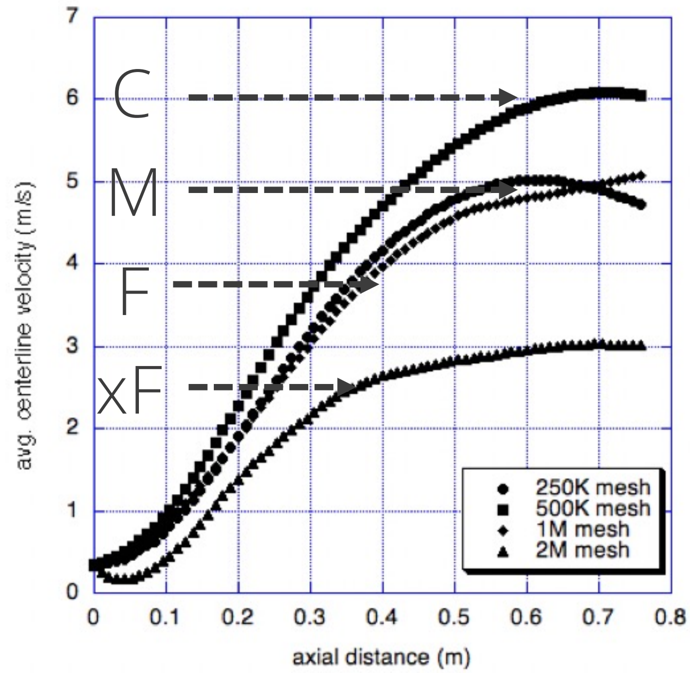


Figure 6. Schematic of the Flame Facility Showing Relationship of Plume, Laser Illumination, and Cameras.



Tieszen et al. “Validation of a Simple Turbulence Model Suitable for Closure of Temporally-Filtered Navier-Stokes Equations Using a Helium Plume”, SAND2005-3210

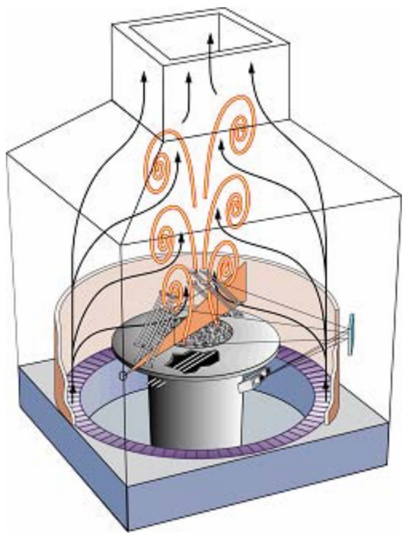
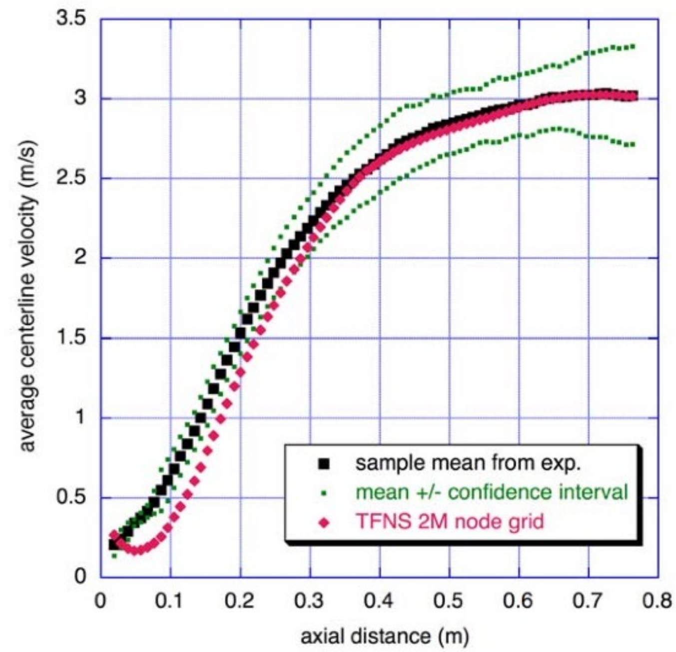


Figure 6. Schematic of the Flame Facility Showing Relationship of Plume, Laser Illumination, and Cameras.



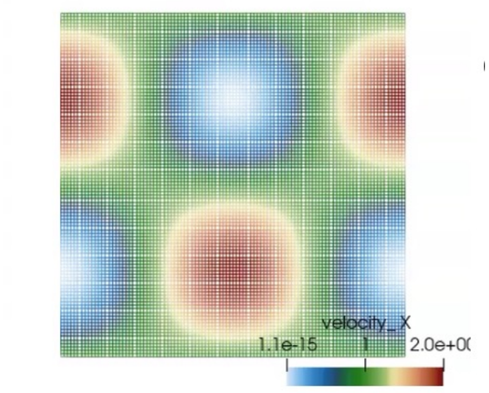
Tieszen et al. "Validation of a Simple Turbulence Model Suitable for Closure of Temporally-Filtered Navier-Stokes Equations Using a Helium Plume", SAND2005-3210

How Code Verification Can Guide Numerical Methods Elucidation

Consistent vs Lumped Mass Matrix

<u>Goal</u>: Explore the time accuracy of the consistent mass matrix approach on a structured mesh

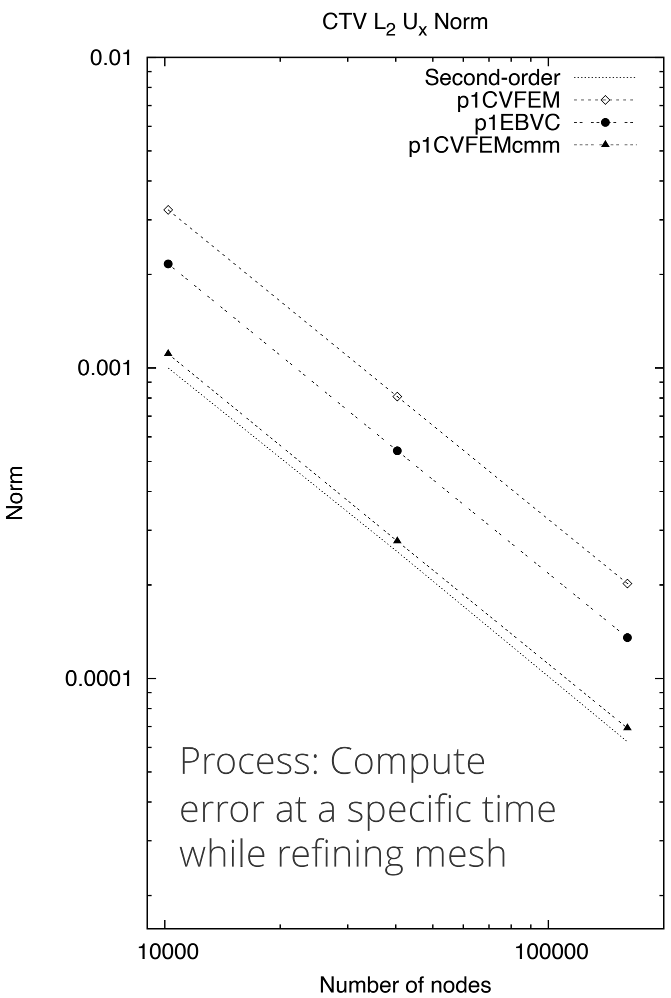
- Convecting Taylor Vortex Case Study: Analytical, transient verification problem



$$u = u_o - cos(\pi(x - u_o t))sin(\pi(y - v_o t))e^{-2.0\omega t}$$

$$v = v_o + sin(\pi(x - u_o t))cos(\pi(y - v_o t))e^{-2.0\omega t}$$

$$p = -\frac{p_o}{4}(cos(2\pi(x - u_o t)) + cos(2\pi(y - v_o t)))e^{-4\omega t}$$



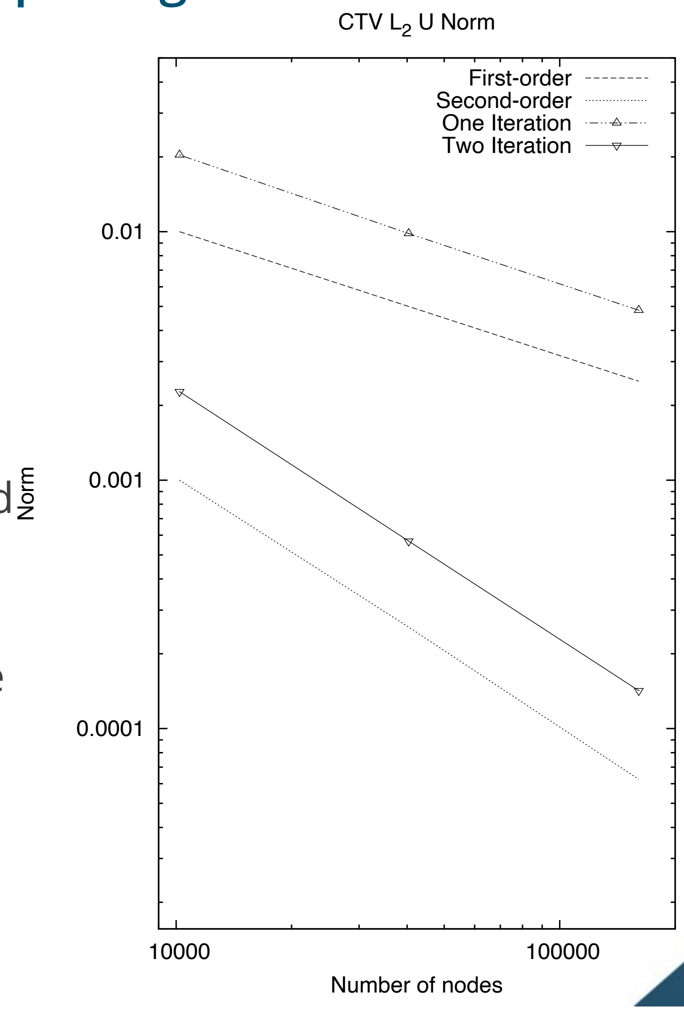
Code or Conceptual Error? Part 1: Time Splitting

<u>Case Study</u>: An Algorithm is thought to be <u>second-order-in-time accurate</u> with one nonlinear iteration: True or False?

- Issa, "Solution of the implicitly discretized fluid flow equations by operator splitting", JCP (1985).
  - Advent of the "Pressure-implicit with Splitting of Operators", or PISO
- PISO is a scheme that defines a series of predictors and correctors in the context of a fully implicit solve

<u>Conclusion</u>?

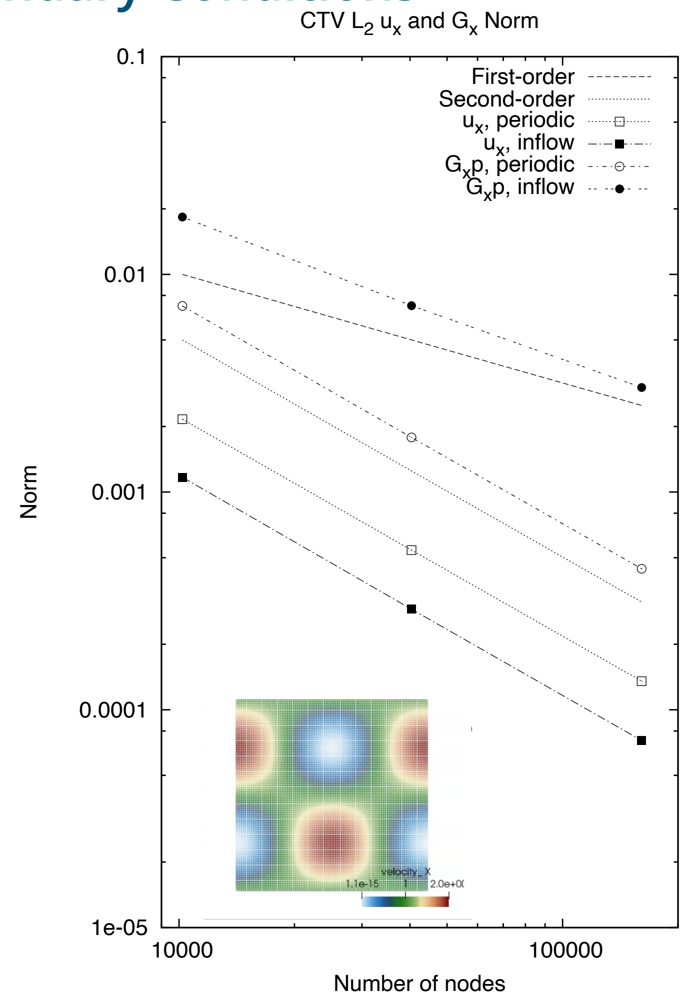
- Sometimes we code a method correctly, however, have a conceptual error in our understanding of whether or not a scheme is design-order accurate when run in the suggested manner



Code or Conceptual Error? Part 2: Boundary Conditions

Case Study: Using the convecting Taylor Vortex case, explore boundary conditions

- Choice 1: Periodic with specified initial condition
  - Requires no formal boundary conditions
- Choice 2: Inflow
  - Easy, velocity is specified as a Dircihlet
- What about pressure? The default approach is to assume a zero normal pressure gradient (no-op) – similar to what would be found at a wall
- A notion of a "spurious numerical boundary layer" (Gresho, 1995) shows a characteristic length $\delta \sim \mathrm{sqrt}(\nu\Delta t)$ error
- We also need to make sure that we are adequately converging the nonlinear system within the time step

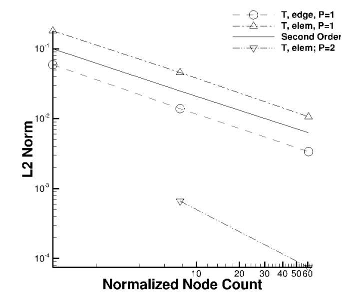


Spatial Verification Process

<u>Process</u>:

1. Start with a mesh of given resolution, Δx_o, or a given total node/cell-centered, etc. count, N
2. Refine the mesh, uniformly, i.e., h-refinement or in polynomial order, p-refinement
3. Run the series of refinement/promoted meshes and compute an <u>integrated</u> norm over the domain, L_∞, L_1, L_2, for example, below shown to be based on a nodal DOF prediction,

$$L_\infty = \sum_i max|\epsilon_i| \quad L_1 = \sum_i \frac{|\epsilon_i|}{N} \quad L_2 = \sum_i \frac{\epsilon_i^2}{N^2}$$



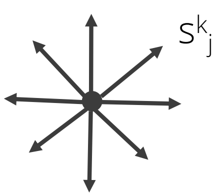
4. Evaluate error relative to what the precise solution think it should be
   - Generally easiest to observe the magnitude of reduction
   - For example, $\Delta x_1 = \frac{\Delta x_o}{2}$, for a first-order in space scheme reduces the error by 2 while for second-order, 4, and third-order, 8, etc.

Lurking, or Hidden Errors…

- In some numerical implementations, an error can exist that is not easily found at a given mesh resolution

For example:

- There may be a boundary error that manifests itself locally at a very small subset of the mesh
- There may be an error that is a driven by the fidelity of the mesh (consider our specified pressure drop in a pipe example)
- Other errors, e.g., for a discrete-ordinance method, we have an underlying *quadrature* error



$$s_j \frac{\partial I}{\partial x_j} + (\mu_a + \mu_s) I = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi}$$

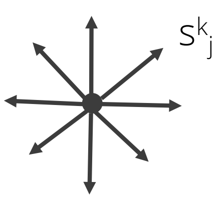
$$s_j^k \frac{\partial I^k}{\partial x_j} + (\mu_a + \mu_s) I^k = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi} \qquad G \approx \sum_k w_k I^k$$

Lurking, or Hidden Errors…

- In some numerical implementations, an error can exist that is not easily found at a given mesh resolution

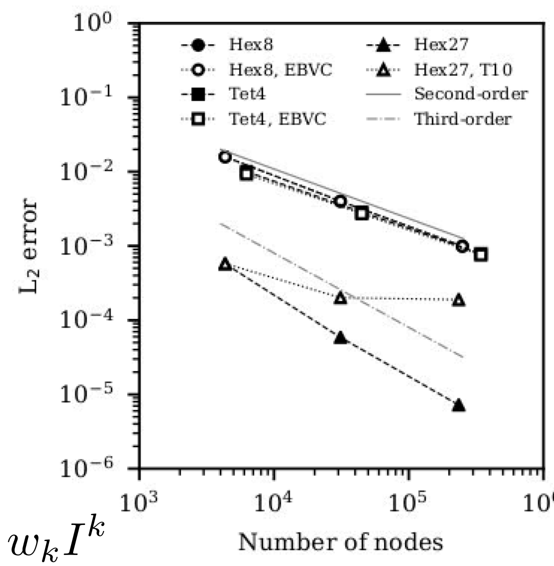
For example:

- There may be a boundary error that manifests itself locally at a very small subset of the mesh
- There may be an error that is a driven by the fidelity of the mesh (consider our specified pressure drop in a pipe example)
- Other errors, e.g., for a discrete-ordinance method, we have an underlying *quadrature* error



$$s_j \frac{\partial I}{\partial x_j} + (\mu_a + \mu_s)\, I = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi}$$

$$s^k_j \frac{\partial I^k}{\partial x_j} + (\mu_a + \mu_s)\, I^k = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi} \qquad G \approx \sum_k w_k I^k$$

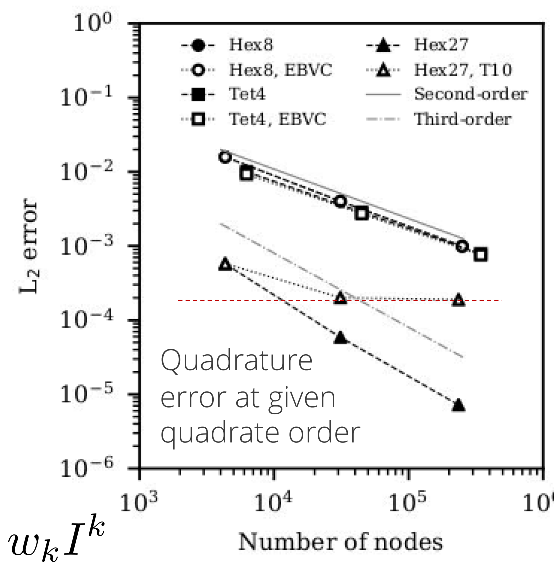


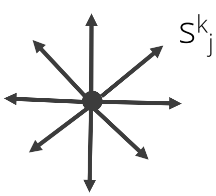
Lurking, or Hidden Errors…

- In some numerical implementations, an error can exist that is not easily found at a given mesh resolution

For example:

- There may be a boundary error that manifests itself locally at a very small subset of the mesh
- There may be an error that is a driven by the fidelity of the mesh (consider our specified pressure drop in a pipe example)
- Other errors, e.g., for a discrete-ordinance method, we have an underlying *quadrature* error



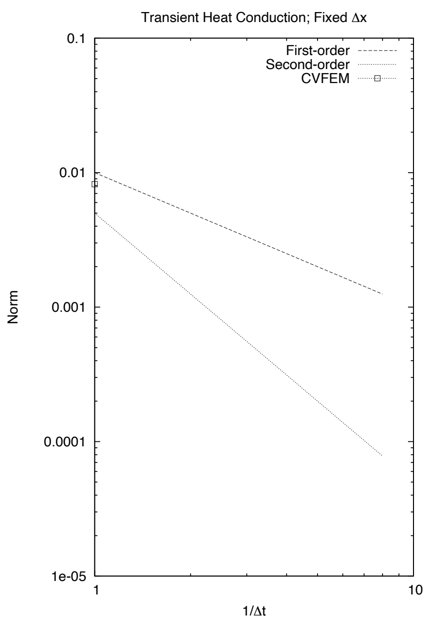


$$s_j \frac{\partial I}{\partial x_j} + (\mu_a + \mu_s) I = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi}$$

$$s^k_j \frac{\partial I^k}{\partial x_j} + (\mu_a + \mu_s) I^k = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi} \qquad G \approx \sum_k w_k I^k$$

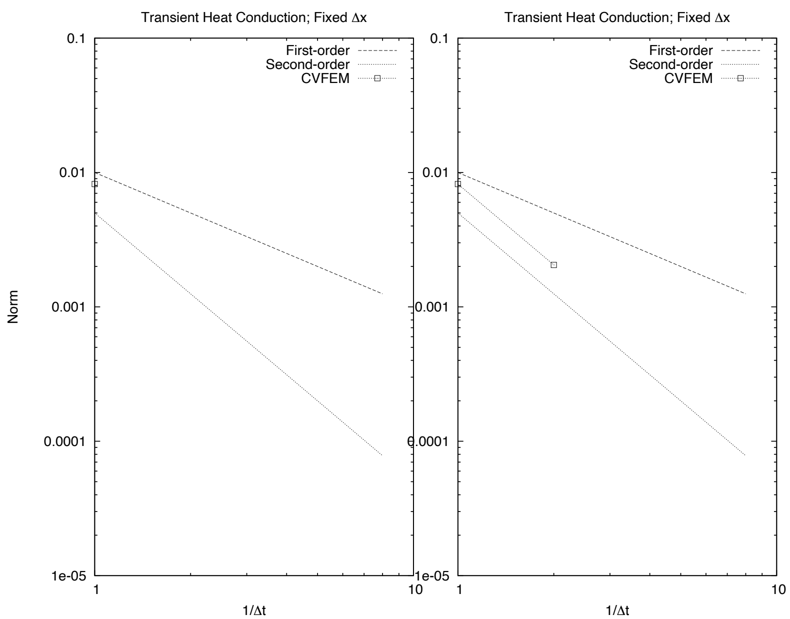
Temporal Verification: Fixed Spatial Resolution; Reduced Timestep

On a given mesh, compute error at a given time: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$



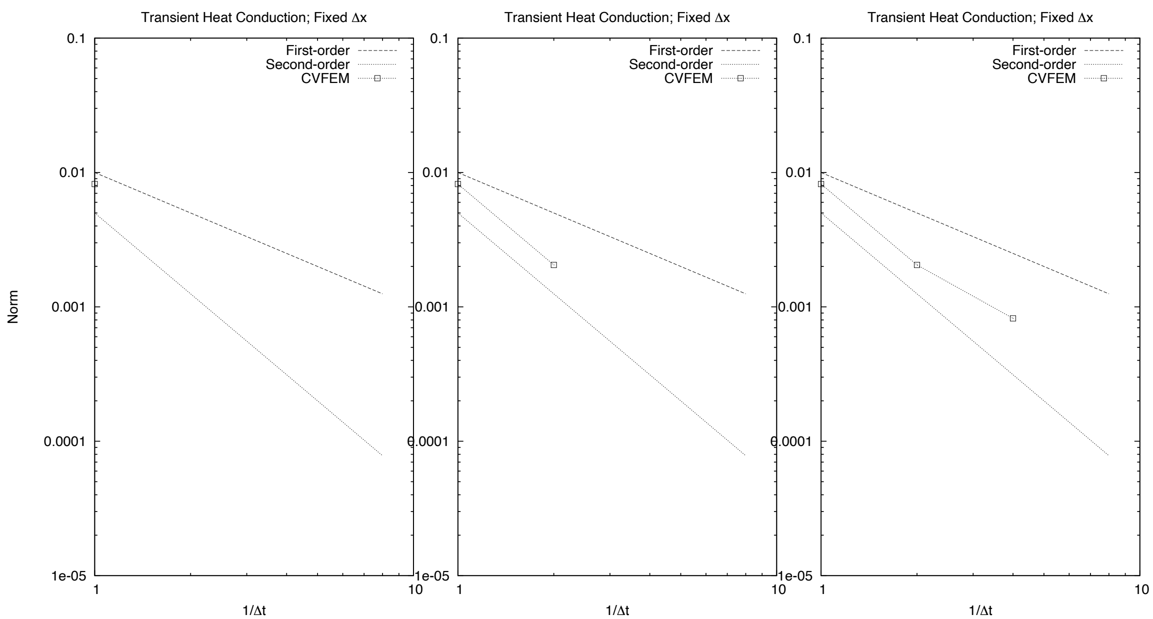
Temporal Verification: Fixed Spatial Resolution; Reduced Timestep

On a given mesh, compute error at a given time: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$



Temporal Verification: Fixed Spatial Resolution; Reduced Timestep

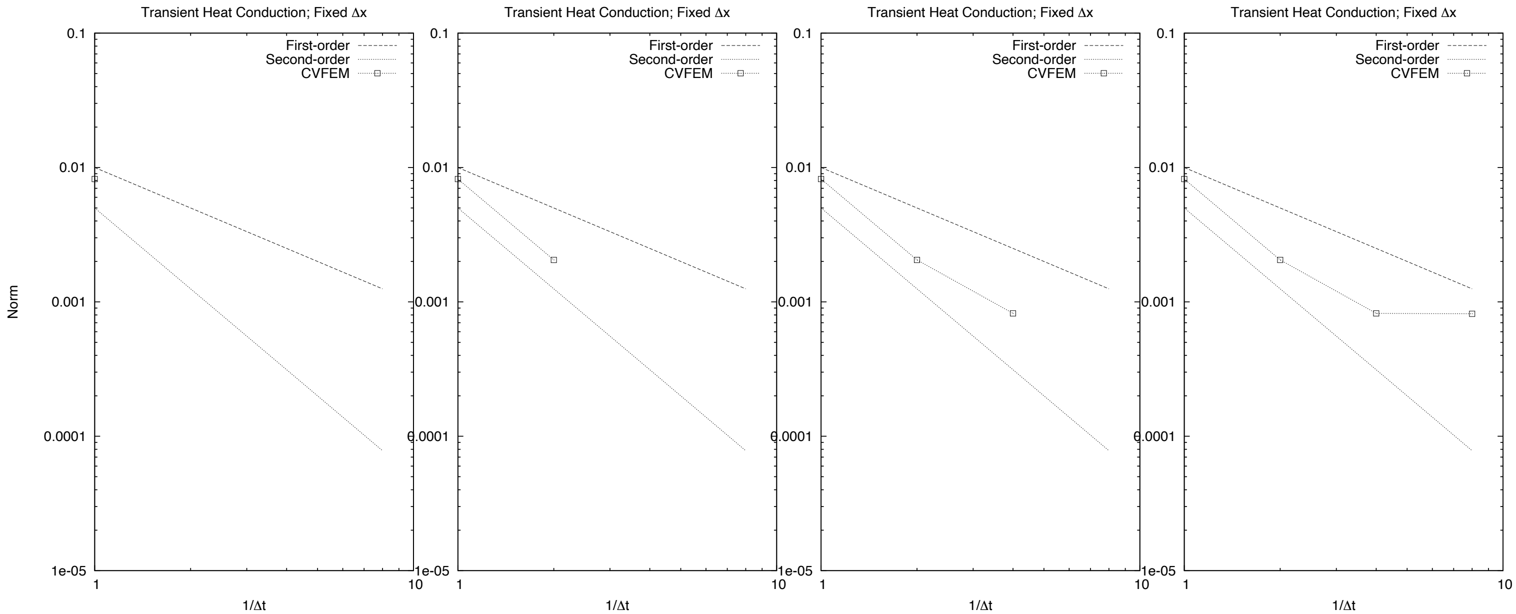
On a given mesh, compute error at a given time: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$



Temporal Verification: Fixed Spatial Resolution; Reduced Timestep

On a given mesh, compute error at a given time: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$

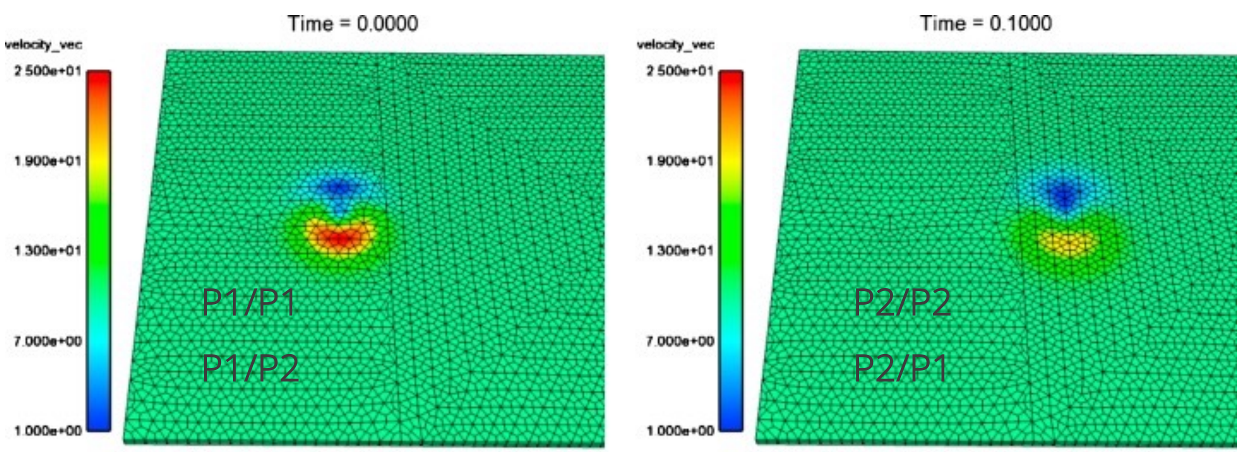
At a given mesh, the mesh spacing is fixed and, eventually, manifests itself



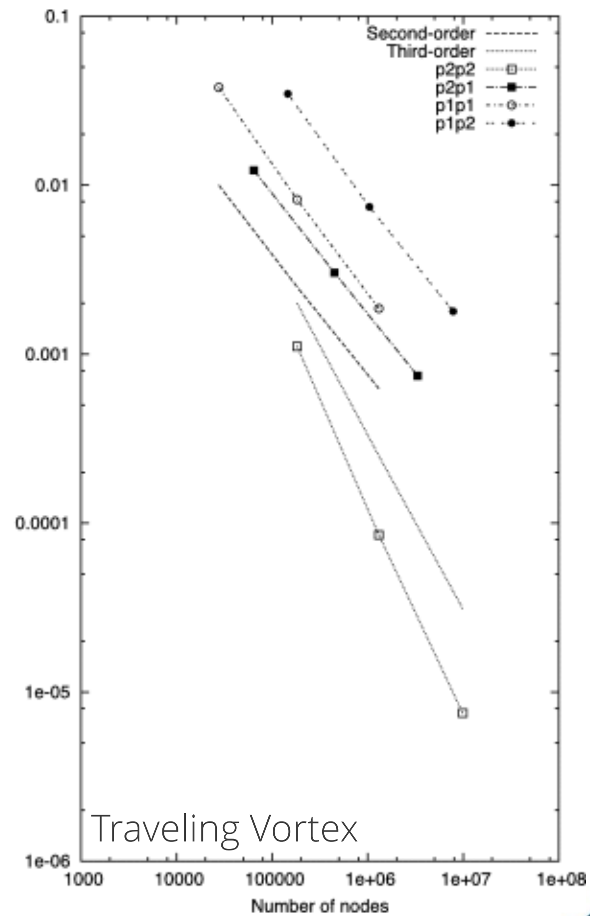
Temporal Verification: Reduction of Timestep as Mesh Is Refined

On a series of uniform mesh refinements, compute error at a given time while reducing time step

- Must ensure that the spatial error does not dominate: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$



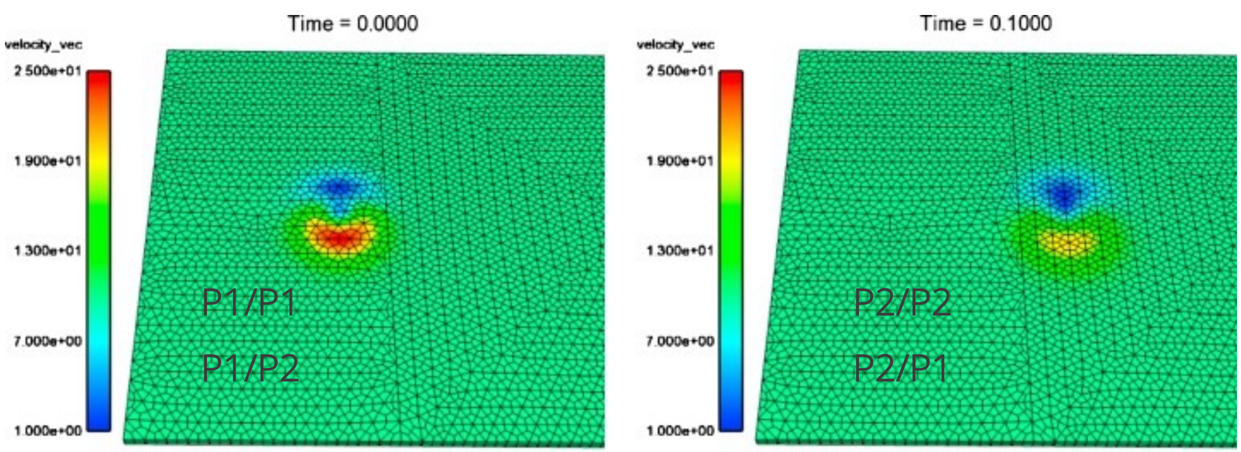
(a) Time = 0.0 seconds.
(b) Time = 0.10 seconds.



Temporal Verification: Reduction of Timestep as Mesh Is Refined

On a series of uniform mesh refinements, compute error at a given time while reducing time step

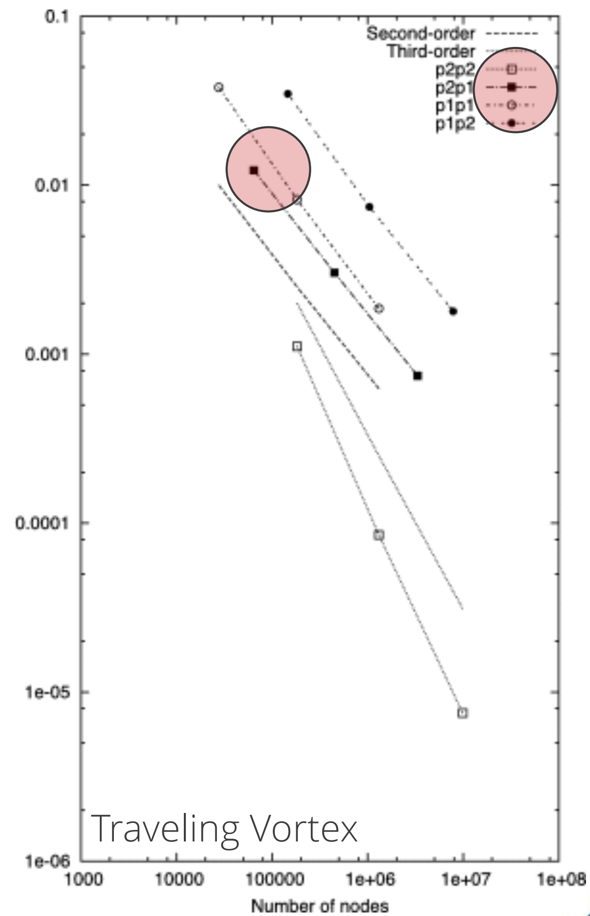
- Must ensure that the spatial error does not dominate: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$



(a) Time = 0.0 seconds.

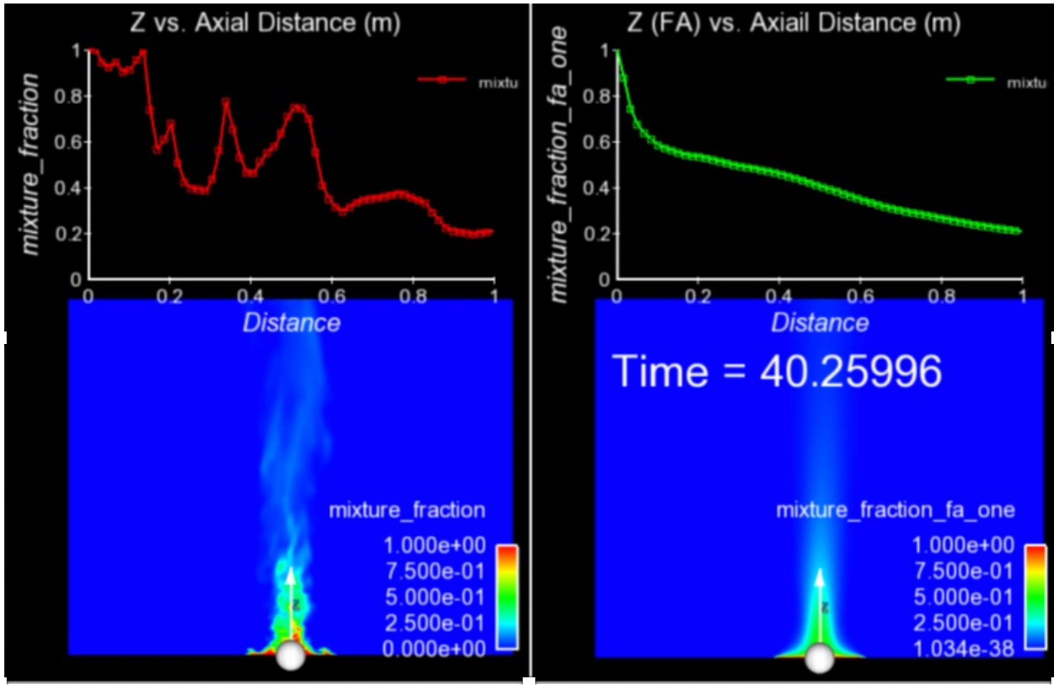
(b) Time = 0.10 seconds.

P2/P1



Finally…. For Transient Flows, Averaging is Required

- The **bane** of turbulent validation: converged statistics require many flow-through times
- Statistical convergence of a given simulation may require many flow-through-times; additional source of uncertainty and/or requirement for quantification of solution convergence



Essentials of Code Verification: Review

<u>Taxonomy</u>: One *verifies* code and *validates* models

- Code verification establishes the numerical accuracy of the underlying discretization for the given partial differential equation set
- Code verification seeks to provide the temporal and spatial accuracy of the underlying discretization approach

For temporal discretization error,

- A two-state Backward Euler time integrator should be first-order in time, specifically the error should scale with Δt
- A three-state BDF2 time integrator should scale with Δt^2
- A multi-stake Runge-Kutta schemes can achieve higher-order accuracy

For spatial discretization error,

- A method is design-order if the observed order of accuracy is Δx^{P+1}, where P is the underlying basis polynomial order